# Investigating Learners' Behaviors and Implementing Intervention in a SPOC

Han Wan, Zihao Zhong, Lina Tang, Xiaopeng Gao
*School of Computer Science and Engineering*
*Beihang University*
Beijing, China
{wanhan, zhongzihao, linatang, gxp}@buaa.edu.cn

***Abstract*—This Work-In-Progress paper is in the Innovative Practice category. In the MOOC-related research field, many researchers analyzed students' learning behavior based on the logging data to predict students' performance and improve the course design. Nowadays, Small Private Online Courses (SPOC) are favored in college education, especially in computing education. This hybrid teaching model allows courses to be conducted through Internet, which enables teachers and students to access the course anytime, anywhere. Besides, multimedia resources, including images, videos, and audio could be contained in course materials to strengthen the expressiveness of SPOC. On the other hand, the online learning management system (LMS) collects all the students' interactions with it. But how could we extract meaningful information from them? And how could we improve the learning outcomes of a SPOC?**

**In this study, we analyzed LMS data from a sophomore Computer Structure course. We applied several data mining techniques and conducted an intervention using several visualization techniques. Features were selected according to Spearman's rank correlation coefficient with grades. The correlation coefficient of these selected features ranged from 0.42 to 0.84. Course data were further processed to predict students' performance. The predicted grade was processed in the form of heatmaps to illustrate students' learning behavior. Besides, we further designed an overall view for teachers' perspective, which contains data of all the students in each heatmap.**

**The predicting models were evaluated by ROC-AUC values. Several hyperparameters were tuned in order to pursue better predict performance. The best ROC-AUC value could reach 97.44%.**

***Keywords—Learning Monitoring; Visualization; Student Performance; Intervention Design; Practice***

## I. INTRODUCTION

Blended learning has become a trend in college education [1]. Introducing online education into traditional courses could increase the diversity of teaching methods and stimulate students' interest in studying through various kinds of intervention techniques such as visualization [2]. These courses are often called Small Private Online Courses (SPOC).

Many of the SPOC courses use Learning Manage Systems (LMS) to assist teaching and learning in online environments. Those systems could greatly reduce the cost for students to access the course resources. With the help of the improvement of the Internet and data mining technologies, we could obtain a huge amount of data from LMS and extract useful information from them. We could also display that information through visualization technology and try conducting intervention based on it.

In this paper, we conducted experiments on a whole-process practice in extracting and analyzing LMS data. We took a sophomore blended course that ran for four consecutive years. Several features were carefully selected from the data through correlation analysis. Based on those features, we built models to predict students' performance (i.e., final grades). The performance of the model was evaluated by ROC-AUC values. Moreover, we tuned several hyperparameters to ensure our models perform better in the datasets. Finally, the models were painted into heatmaps using visualization techniques to assist intervention on students.

## II. BACKGROUND

LMS enables teachers to collect various information such as the interaction between students and the system. Those data could be utilized in much more fine-grained course analyses. Given its importance, previous research has focused on studying the ways to better using LMS data.

Conijn, Van den Beemt, and Cuijpers [3] conducted a research on a graduate-level blended MOOC. They predicted students' performance by correlation analysis and multiple regression.

Liao et al. [4] collected clicker data from a course to identify poorly performed students. The data was fed into an SVM model for prediction. This research was conducted across two consecutive terms, using the former one's data to train and the latter one's data to test.

Yang and Li [5] performed a two-year study on predicting high school students' performance. The tools they have used included Student Attribute Matrix (SAM) and neural networks.

Most of the related researches, however, did not make full use of the data mining techniques which could probably further enhance the performance. None of the above researches performed feature engineering. Research [5] did not mention hyperparameter optimization though it was commonly used in neural networks. These motivated us to conduct a more thorough data analysis.

Based on the data extracted from LMS, we could conduct interventions on students. Various intervention methods are possible in blended courses based on their online parts. For example, the visualization techniques enable teachers to generate graphs or videos to illustrate students' learning status in a prettier way. Moreover, LMS enables personalized intervention for hundreds or even more students.

Previous research utilized multiple visualization techniques in education. Auvinen, Hakulinen, and Malmi [6] introduced achievement badges, a kind of gamification, into their courses. Moreover, they selected several variables related to badges to draw heatmaps for students. Podgorelec and Kuhar [7] used their experience in commercial environments to create dashboards for instructors of online courses. They developed various kinds of panels, showing students' performance on different parts of the course. Wang,

Peng, Cheng, Zhou, and Liu [8] developed a knowledge visualization system based on a Java-programming course. They drew knowledge maps as well as other visualized graphs on Java development.

During the COVID-19 pandemic, Midak, Kravets, Kuzyshyn, Baziuk, and Buzhdyhan [9] applied Augmented Reality (AR) in a chemical course. They produced 3D images of molecules to give students a better understanding of chemical compounds' structure. Dengal [10] chose to use Virtual Reality (VR) technologies to visualize finite state machines.

Garay, Tchernykh, Drozdov, Garichev, Nesmachnow, and Torres-Martinez [11] used simulation visualization in an undergraduate computer science course. They graphically illustrate the concepts of a VHDL-based system for the ease of students to understand.

## III. Methodology

### A. Course Background

The data source of this research was a sophomore Computer Structure course that ran from Fall Semester 2017 to Fall Semester 2020. The course was conducted by blended teaching methods. There were slight modifications between different rounds of the course. In the traditional classroom, students mainly learned the theoretical part of the course. Meanwhile, in the online course platform, students could study the experimental part by themselves. Students were required to design a 5-stage pipeline CPU based on MIPS architecture using Verilog HDL to pass the course.

The online course platform was developed based on the Open edX platform. It offered various types of learning resources, including textual materials or videos for students to learn. There was a course forum where students and teachers could freely discuss course problems. Moreover, we developed an automated grading system based on the edX XQueue interface. This system helped students judge the correctness of their CPUs conveniently.

During the interaction with the students, the course platform could automatically save logs of various types. When a student watched a video, the platform could record events like playing video, stopping video, and seeking video. When a student submitted a problem, the platform could record the correctness of the submission, as well as events such as clicking 'show answer'. When a student discussed in the course forum, the platform could record creating, reading, and replying thread events. The raw data were further processed and analyzed, generating the features used in this study.

### B. Feature Engineering

We first manually selected features in TABLE I. Those features should be able to be expressed in a one-dimensional variable for the sake of simplicity in visualization and correlation analysis. The features should be relatively easy for students to improve in order to conduct effective interventions. The ranges of values of the features might vary between different semesters. Therefore, we performed normalization on features in each semester, mapping them to the range $[0,1]$.

We further selected features by calculating the correlation coefficient between features and the final grades of the students. As the features usually do not follow any regular distributions, we utilized Spearman's rank correlation coefficient, which measures the relationship only by the rank of features and grades.

TABLE I. Manually Selected Features and Their Meanings

| Feature | Meaning |
|---|---|
| Submission AC number | How much correct submissions a student has made. |
| Forum activities | A composite feature generally considers sending, replying, and reading a thread. |
| Total duration | The total time a student has spent in course study. |
| Total courseware access | The total number a student has accessed the course platform. |
| Number distinct problem submit | How many distinct problems a student has submitted. |
| Observed event variance | The variance of a student's events per day. |
| Observed problem duration | The average time a student uses in solving problems. |
| Average number submissions | The average number of submissions a student needs to pass a problem. |
| Total video duration | The total time a student watches videos. |

### C. Visualized Intervention with Heatmaps

Using the selected features, we were able to visualize and illustrate them to students and teachers. Our visualization method was generating heatmaps for every two features, where the x-axis and the y-axis represent the two features, respectively. The value of a specific point should be the prediction of students whose corresponding features have exactly the value of this point. Formally, assuming the features range between $[0,1]$ and grades range in $[0,100]$, the heatmap could be represented as a function $f: [0,1]^2 \to [0,100]$. We should make a precise enough prediction to conduct accurate interventions.

In this problem, however, it is impossible to predict for every point since there are infinitely many of them. As a reasonable approximation, we divided $[0,1]^2$ into $B^2$ equal square blocks, which means dividing both axes into $B$ equal blocks. $B$ should be a large enough integer to guarantee the precision of prediction and the beautifulness of heatmaps.

Each block was represented by its central point. For each point (i.e., the central point of each block), we selected $k$ nearest neighbors from the training set and predicted its corresponding grade as the average of the $k$ points. In this study, data from the first three semesters were regarded as the training set. The use of $k$ nearest neighbors helped handle missing values and reduce the effect of outliers [6].

We applied a brute force algorithm to find the $k$ nearest neighbors. Assuming there are $n$ points in the training set, we could generate a list containing the distance between the $n$ points and the selected point. Then sort the list and pick up the first $k$ element to calculate the answer. The time complexity of the algorithm is $O(B^2 n \log n)$. As $B$ and $n$ in our context ranged from a few hundred to a few thousand, the brute force algorithm was acceptable. For larger datasets, advanced data structures such as k-d tree could be utilized to achieve better time complexity.

### D. Model Optimization

The previous sections involved some hyperparameters. For example, feature 'Forum activities' is a combination of creating thread, replying thread, and reading thread.

Apparently, those three events should have different contributions. Creating thread and replying thread should contribute more than reading thread. Therefore, we defined 'Forum activities' as a linear combination of the number of the three events and assigned different coefficients to them. Without loss of generality, we set the coefficient of reading thread to 1. The coefficients of the other two events were regarded as hyperparameters. We selected several empirical values for each coefficient and did grid search to find parameters giving the highest average correlation coefficient.

$B$ and $k$ in the model could also be regarded as hyperparameters. As the use of $k$ nearest neighbor decreased the variation of predicted grades, it was hard to use Minimum Squared Error to evaluate the model. As another approach, we set the target of prediction to whether the student passed the course, which defined the problem as a binary classification problem. To examine the performance of the prediction, we checked the ROC-AUC value of it. Again, we chose several empirical values of $B$ and $k$ and performed grid search on them. The $B$ and $k$, which led to the highest ROC-AUC value, were chosen as the final hyperparameters in prediction.

## IV. Results and Discussion

### A. Feature Engineering

Based on historical data from Fall Semester 2017 to Fall Semester 2019, we calculated Spearman's rank correlation coefficient between features and grades for each semester separately, as shown in TABLE II. The five features with the highest correlation coefficient of each semester were marked in bold. We could conclude that 'Submission AC number', 'Forum activities', 'Total duration', 'Total courseware access', and 'Number distinct problem submit' were the most related features.

TABLE II. Correlation Coefficients between Features and Grades

| Feature Name | Semester | | |
|---|---|---|---|
| | *Fall 2017* | *Fall 2018* | *Fall 2019* |
| **Submission AC number** | **0.8366**$^{***}$ | **0.7686**$^{***}$ | **0.8284**$^{***}$ |
| **Forum activities** | **0.6840**$^{***}$ | **0.5192**$^{***}$ | **0.5026**$^{***}$ |
| **Total duration** | **0.6279**$^{***}$ | **0.4765**$^{***}$ | **0.5962**$^{***}$ |
| **Total courseware access** | **0.6061**$^{***}$ | **0.4218**$^{***}$ | **0.5534**$^{***}$ |
| **Number distinct problem submit** | **0.6016**$^{***}$ | **0.6637**$^{***}$ | **0.6687**$^{***}$ |
| Observed event variance | 0.4325$^{***}$ | 0.1842$^{***}$ | 0.1614$^{**}$ |
| Observed problem duration | 0.3852$^{***}$ | 0.2279$^{***}$ | 0.4670$^{***}$ |
| Average number submissions | 0.1290$^{*}$ | -0.0152 | 0.2915$^{***}$ |
| Total video duration | -0.1829$^{***}$ | -0.0550 | 0.1146$^{*}$ |

[a.] *p<.05. **p<.01. ***p<.001.

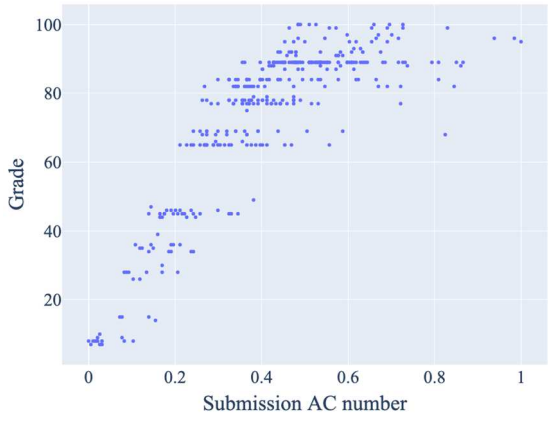


Fig. 1. Scatter plot of feature 'Submission AC number' and grades

To examine the relationship between the selected features and student grades in another perspective, we plotted them on scatter plots such as Fig. 1. It was apparent that there was a positive relationship between feature 'Submission AC number' and grades of students.

### B. Hyperparameter Optimization

As stated in section III.D, we optimized some hyperparameters before utilizing them in the model. The experimental result on forum activity coefficients was shown in TABLE III. Therefore, the coefficients of sending thread, replying thread, and reading thread were set to 50, 150, and 1, respectively. This result indicated that the weight of sending or replying thread was much higher than the one of reading thread.

An example showing the ROC-AUC values of models with different $B$s and $k$s was listed in bThe models were predicted using features 'Submission AC number' and 'Forum activities'. Therefore, we set $B = 150$ and $k = 200$ to achieve the best predict performance. The ROC-AUC value reached 97.44% under the hyperparameters, which was a rather high value.

TABLE III. Average Correlation Coefficient on Different Parameters

| Reply Coefficient | Send Coefficient | | | | | | |
|---|---|---|---|---|---|---|---|
| | *1* | *5* | *20* | *50* | *100* | *150* | *200* |
| 1 | 4982 | 5101 | 5216 | 5555 | 5631 | 5654 | 5650 |
| 5 | 5046 | 5154 | 5254 | 5568 | 5639 | 5660 | 5654 |
| 20 | 5104 | 5212 | 5294 | 5586 | 5649 | 5669 | 5661 |
| 50 | 5355 | 5404 | 5450 | 5624 | 5680 | **5686** | 5682 |
| 100 | 5397 | 5432 | 5464 | 5609 | 5669 | 5680 | 5678 |
| 150 | 5379 | 5404 | 5433 | 5565 | 5632 | 5657 | 5661 |
| 200 | 5339 | 5366 | 5393 | 5524 | 5593 | 5622 | 5637 |

[b.] The value in the table is 10,000 times the real value

[c.] All of the p was less than .001.

TABLE IV. ROC-AUC on Different Parameters

| *B* | *k* | | | | |
|---|---|---|---|---|---|
| | *1* | *50* | *100* | *150* | *200* |
| 100 | 0.9131 | 0.9711 | 0.9737 | 0.9742 | 0.9737 |
| 125 | 0.9193 | 0.9720 | 0.9736 | 0.9739 | 0.9739 |
| 150 | 0.9182 | 0.9736 | 0.9737 | 0.9740 | **0.9744** |
| 175 | 0.9311 | 0.9720 | 0.9738 | 0.9736 | 0.9742 |
| 200 | 0.9339 | 0.9717 | 0.9738 | 0.9741 | 0.9740 |

### C. Visualized Intervention with Heatmaps

Based on the historical data from Fall Semester 2017 to Fall Semester 2019 and the optimized parameters, we could generate the prediction of each pair of features. We could further draw heatmaps base on this prediction. Each of the $B^2$ blocks were assigned a color according to the predicted grade of its central point. In our implementation, we used red to represent low grades and used yellow to represent high grades. For students' view, we used a scatter point to show his current normalized feature value. Therefore, he could realize which feature he did not perform well and would be alerted to improve this kind of feature.

Fig. 2 was an example heatmap of a student from Fall Semester 2020. The two features were 'Submission AC number' and 'Forum activities'. This student had passed a relatively high number of problems. However, he did not participate in the course forum very often. He could further improve his performance by sending or replying to more threads in the forum. Though we set the color of the point to

the predicted grade, its exact value was not displayed to students. We encouraged students to focus on which part of the course they could improve instead of how many points they could exactly get in this course.

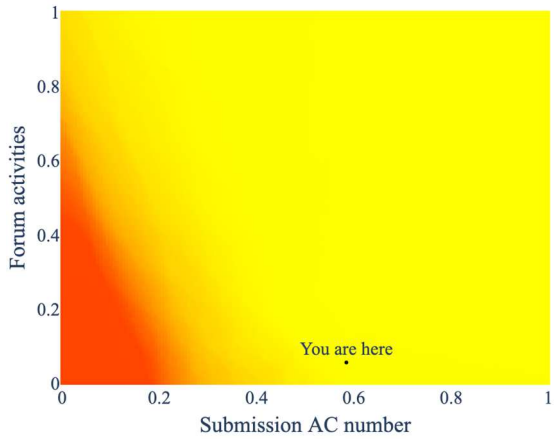


Fig. 2. Heatmap of features' Submission AC number' and 'Forum activities' (students' view)

For teachers' view, we plotted points of all students so that teachers were able to grasp the overall performance of students. As Fig. 3 showed, a large number of students did not talk in the forum very often. Besides, all students who fell in the red area had 'Forum activities' less than 0.2. It was possibly useful to encourage those students who had difficulties in solving problems to request help in the course forum. Notably, the variation of predicted grades tended to be smaller than its real value. This phenomenon could be easily explained by the process of taking average among the $k$ nearest neighbors.

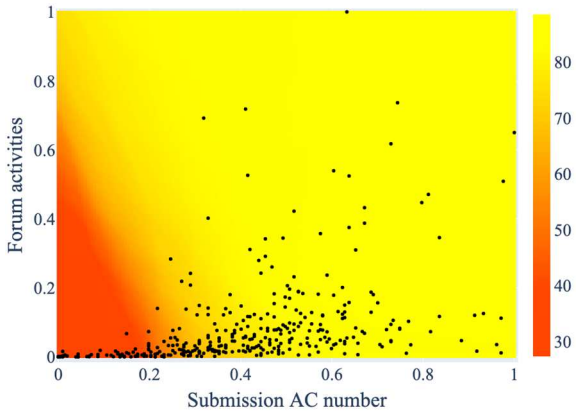


Fig. 3. Heatmap of features' Submission AC number' and 'Forum activities' (teachers' view)

TABLE V. EXAMPLE OF STUDENTS' FEATURE VALUES AND PREDICTED GRADES

| Student ID | Submission AC number | Forum activities | Predicted grade |
|---|---|---|---|
| 1918***8 | 0.116 | 0.005 | 27.415 |
| 1937***4 | 0.106 | 0.005 | 27.415 |
| 1918***1 | 0.087 | 0.001 | 27.415 |
| 1937***4 | 0.010 | 0.016 | 27.455 |
| 1923***7 | 0.000 | 0.001 | 27.465 |

We also offered a table that listed students according to the increasing order of their performance, such as TABLE V. This table allowed teachers to find out the worst performed students and pay special attention to them. We could see from the table that those five students had solved very few problems and rarely participated in forum talks.

## V. CONCLUSION

In this paper, we proposed a systematic method of analyzing LMS data and visualizing them by heatmaps. This method could help identify at-risk students and conduct interventions on them. The course platform, which recorded every interaction between it and the students, provided a huge amount of raw data to analyze. We extracted nine features from the data and further selected five of them by evaluating Spearman's rank correlation coefficient. This process guaranteed that those selected features were suitable to represent the learning status of students. We illustrated the model by drawing heatmaps with colors that represent predicted grades. The contrast of different colors could give students intuition on whether they had done good or bad. As this intervention contained multiple dimensions of features, students could easily find out their weaknesses. Finally, the optimization of hyperparameters forum coefficients, as well as $B$ and $k$, helped the model reach a better performance.

## VI. FUTURE WORKS

This research is still working in progress. There are several aspects where we could further study in the future. We would like to integrate the graphs into our open edX-based course platform. Besides, the graphs could be directly pushed to students through WeChat (a messaging app in China) platform or e-mails to ensure the students could see them.

Based on the above techniques, we planned to set up a controlled experiment by displaying the graphs to a part of the students while hiding them from the other ones. This experiment might help in estimating how much this method benefits students in real learning environments.

More other kinds of visualization techniques could be utilized in our intervention. As an example, radar charts could display all the features in a single graph. It would be relatively simple to draw such graphs since we have already normalized the feature values. However, it might be hard to combine the radar graphs of multiple students (for teachers' view), which could be further discussed.

It could also be beneficial if we offer a student animated GIFs containing his/her historical graphs. By viewing the changes in the graphs, students could feel their progress and be encouraged to make better efforts in the future.

## ACKNOWLEDGMENT

This work was partially supported by the National Natural Science Foundation of China (61907002) and by the Computer Education Research Association of Chinese Universities (CERACU2019R12).

## REFERENCES

[1] Lu, O. H., Huang, A. Y., Huang, J. C., Lin, A. J., Ogata, H., & Yang, S. J. (2018). Applying learning analytics for the early prediction of Students' academic performance in blended learning. *Journal of Educational Technology & Society*, *21*(2), 220-232.

[2] Jin, S. H. (2017). Using visualization to motivate student participation in collaborative online learning environments. *Journal of Educational Technology & Society*, *20*(2), 51-62.

[3] Conijn, R., Van den Beemt, A., & Cuijpers, P. (2018). Predicting student performance in a blended MOOC. *Journal of Computer Assisted Learning*, *34*(5), 615-628.

[4] Liao, S. N., Zingaro, D., Thai, K., Alvarado, C., Griswold, W. G., & Porter, L. (2019). A robust machine learning technique to predict low-performing students. *ACM Transactions on Computing Education (TOCE)*, *19*(3), 1-19.

[5] Yang, F., & Li, F. W. (2018). Study on student performance estimation, student progress analysis, and student potential prediction based on data mining. *Computers & Education*, *123*, 97-108.

[6] Auvinen, T., Hakulinen, L., & Malmi, L. (2015). Increasing students' awareness of their behavior in online learning environments with visualizations and achievement badges. *IEEE Transactions on Learning Technologies*, *8*(3), 261-273.

[7] Podgorelec, V., & Kuhar, S. (2011). Taking advantage of education data: Advanced data analysis and reporting in virtual learning environments. Elektronika ir Elektrotechnika, 114(8), 111-116.

[8] Wang, M., Peng, J., Cheng, B., Zhou, H., & Liu, J. (2011). Knowledge visualization for self-regulated learning. Journal of Educational Technology & Society, 14(3), 28-42.

[9] Midak, L. Y., Kravets, I. V., Kuzyshyn, O. V., Baziuk, L. V., & Buzhdyhan, K. V. (2021, March). Specifics of using image visualization within education of the upcoming chemistry teachers with augmented reality technology. In Journal of Physics: Conference Series (Vol. 1840, No. 1, p. 012013). IOP Publishing.

[10] Dengel, A. (2018, December). Seeking the treasures of theoretical computer science education: Towards educational virtual reality for the visualization of finite state machines. In 2018 IEEE international conference on teaching, assessment, and learning for engineering (TALE) (pp. 1107-1112). IEEE.

[11] Garay, G. R., Tchernykh, A., Drozdov, A. Y., Garichev, S. N., Nesmachnow, S., & Torres-Martinez, M. (2019). Visualization of VHDL-based simulations as a pedagogical tool for supporting computer science education. Journal of Computational Science, 36, 100652.